

Project Proposal Report on

SoloSphere: A Web-Based Freelancer Hiring System

Submitted By

Student Name	Student ID
Alauddin	0242310005101415
Md Rubait Ahmed Sifat	0242310005101837

Semester: Spring 25

Course Title: Software Engineering

Course Code: CSE315

Submitted To:

Pranto Protim Choudhury

Designation: Lecturer

Department of Computer Science and Engineering

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

December 12, 2025

Introduction

SoloSphere is a web-based freelancer hiring platform that connects clients with skilled professionals in an organized and efficient way. The system allows users to post jobs, explore available opportunities, and hire freelancers across various service categories. Clients can easily manage job listings and review bids, while freelancers can submit proposals and track their applications.

In today's digital environment, finding reliable talent or suitable work can be challenging. SoloSphere simplifies this process by providing a secure and user-friendly marketplace that supports clear communication, transparent bidding, and smooth collaboration. The platform helps users build professional connections, complete projects effectively, and access new earning or hiring opportunities with confidence.

Objective

- **Make Hiring Easy** – Enable clients to quickly post jobs and find suitable freelancers without complications.
- **Connect Talent and Opportunities** – Provide freelancers with a platform to discover projects that match their skills.
- **Simplify Job Management** – Offer tools for organizing job posts, tracking bids, and managing work efficiently.
- **Ensure Fair & Transparent Bidding** – Allow freelancers to submit proposals and compete openly with clear job details.
- **Promote Secure Collaboration** – Create a trusted environment where both clients and freelancers can interact safely.
- **Support Efficient Communication** – Help users stay updated with bid status and job progress through smooth interaction.
- **Improve Workflow for Users** – Make applying, hiring, and updating job information fast and hassle-free.

- **Build a Professional Community** – Encourage meaningful connections between clients and freelancers across multiple categories.
- **Turn Data into Decision Power** – Provide insights through job status, bid counts, and user activity to support informed hiring.

Key Features

- **user Registration and Login**

A simple and secure sign-up process that allows clients and freelancers to create accounts quickly. The system also supports safe login, ensuring that users can access their profiles and job-related activities without any hassle.

- **Personalized User Dashboard**

A clean, user-friendly dashboard that displays essential information such as posted jobs, applied jobs, bid requests, and recent interactions. Both clients and freelancers get a customized view tailored to their activities.

- **Job Posting and Management**

Clients can easily create new job listings by adding details such as title, description, budget, deadline, and category. They can also edit or delete their posted jobs whenever necessary, making project management simple and flexible.

- **Freelancer Job Search and Filters**

Freelancers can browse all available jobs, search by keywords, and filter by categories. This helps them quickly find opportunities that match their skills and preferences.

- **Bid Submission System**

Freelancers can submit bids on jobs with custom proposals, expected delivery time, and pricing. The system prevents duplicate bids and ensures fair participation for all applicants.

- **Bid Review and Approval**

Clients receive all bid requests on their posted jobs and can review, accept, or reject proposals based on freelancer profiles and bid details. This feature ensures transparent and efficient hiring decisions.

- **My Bids Tracking**

Freelancers can monitor the status of all their bids—whether they are pending, accepted, or rejected. This helps them stay updated and manage their workflow effectively.

- **Real-Time Status Updates**

Bid status changes such as “Approved” or “Rejected” are reflected instantly in the system, ensuring that freelancers are always aware of their project progress.

- **Job Details View**

Each job has a dedicated detail page showing the full description, budget, required skills, buyer information, and current bid count. Freelancers can easily understand requirements before applying.

- **Search and Category-Based Filtering**

Advanced search functionality allows users to find jobs using keywords, while category filters help narrow down job lists for more relevant browsing. This improves the discovery experience for freelancers.

- **Secure and Reliable System**

The platform ensures secure handling of user information through authentication, authorization, and protected routes. Sensitive actions like job editing, bidding, and bid approval are safeguarded.

- **Seamless Communication Workflow**

Although messaging is not yet implemented, the existing job/bid status system allows smooth coordination between clients and freelancers through transparent updates and structured interactions.

- **Scalable and Organized Backend Structure**

The backend supports structured data handling using Node.js, Express, and MongoDB, ensuring high performance, quick queries, and smooth operations even with large numbers of users and jobs.

Scenario Writing

Scenario writing for SoloSphere describes how clients and freelancers interact with the platform to post jobs, submit bids, manage proposals, and complete hiring tasks smoothly.

These step-by-step use cases show how the system supports users in practical, real-life workflows.

Scenario-1: User Dashboard Access

Scenario Description:

- A user logs into the system using valid credentials.
- The dashboard displays personalized information such as posted jobs, received bids, or submitted proposals.
- The user checks new notifications and reviews recent updates related to job activities.
- They navigate to different sections such as “My Bids” or “My Posted Jobs.”
- Dashboard interaction completed successfully.

Scenario-2: Job Posting by Client

Scenario Description:

- A client selects the “Post a Job” option from the dashboard.
- They fill in job details such as title, category, budget, and description.
- The client submits the form, and the system stores the job in the database.
- The job immediately appears in the “All Jobs” list and under the client’s “My Posted Jobs” section.
- Job posting completed successfully.

Scenario-3: Freelancers Browsing and Searching Jobs

Scenario Description:

- A freelancer navigates to the “All Jobs” page.
- They browse through available job listings and apply search keywords such as “logo design” or filter by category.
- The system displays matching results based on the search terms and filters.
- The freelancer selects a job to view full details before applying.
- Job search and filtering completed successfully.

Scenario-4: Bid Submission by Freelancer

Scenario Description:

- A freelancer opens a job detail page they want to apply for.
- They submit a bid including proposed price, delivery time, and a short proposal message.
- The system checks whether the user has already bid on this job.
- If not, the bid is submitted and the job's bid count is updated.
- Bid submission completed successfully.

Scenario-5:Client Reviewing Bid Requests

Scenario Description:

- A client opens the “Bid Requests” page to review bids received on their posted jobs.
- The system displays all proposals submitted by freelancers along with their details.
- The client evaluates each bid and selects the most suitable freelancer.
- They update the bid status to “Accepted” or “Rejected.”
- Bid review and decision-making completed successfully.

Scenario-6: Freelancer Tracking Bid Status

Scenario Description:

- A freelancer visits the “My Bids” section to check updates on their submissions.
- The system shows whether a client has accepted, rejected, or not yet responded to each bid.
- The freelancer monitors their progress and decides which jobs to focus on next.
- Bid status tracking completed successfully.

Scenario-7:Updating an Existing Job

Scenario Description:

- A client wants to modify a job they previously posted
- They open the “My Posted Jobs” page and select a job to update.
- The system loads the current job details into an editable form.
- The client updates the information and saves changes.
- The updated job information is stored in the database and reflected instantly.
- Job update completed successfully.

Scenario-8: Job Deletion

Scenario Description:

- A client decides to remove a job that is no longer needed.
- From “My Posted Jobs,” they click the delete button on a job listing.
- The system confirms the action and removes the job from the database.
- The job disappears from both the general listing and the client’s own dashboard.
- Job deletion completed successfully.

Scenario-9: Real-Time System Notifications

Scenario Description:

- A freelancer receives a notification that their bid status has been updated.
- A client receives a notification when a new bid is submitted on their posted job.
- Users check their dashboard to view these updates.
- Notifications help users stay informed about critical actions.
- System notification handling completed successfully.

Each scenario focuses on key interactions such as posting jobs, submitting bids, reviewing proposals, updating job details, and tracking activity—showcasing how SoloSphere makes freelance hiring simple, structured, and efficient.

Stakeholders

Distinct stakeholders involved in the SoloSphere system, with their specific roles:

1. Clients (Job Posters / Buyers)

- **Role:** Clients are the users who create and manage job postings on the platform. They provide project details, set budgets, review freelancer bids, and select the best candidate for their task.
- **Key Interaction:** Clients use the system to create job listings, monitor incoming bids, compare proposals, manage posted jobs, and finalize hiring decisions.

2. Freelancers (Service Providers / Job Seekers)

- **Role:** Freelancers are the users who browse available jobs, submit bids, and offer their services. They compete for projects based on skills, pricing, and proposal quality.

- **Key Interaction:** Freelancers use the system to search jobs, apply with custom proposals, track bid status, manage accepted tasks, and explore new opportunities.

3. System Administrator

- **Role:** Administrators oversee the platform to ensure smooth operation, data accuracy, security, and user compliance. They handle system maintenance, user management, and content moderation.
- **Key Interaction:** Administrators use the system to monitor activities, manage user accounts, resolve disputes, remove inappropriate content, and ensure platform stability.

4. . Notification & System Services (Internal Stakeholder)

- **Role:** Handles automated notifications such as bid updates, job approval alerts, and status changes. Ensures real-time communication between clients and freelancers.
- **Key Interaction:** Sends alerts to users for new bids, bid decisions, job updates, and important system events to maintain smooth interaction.

5. Future Payment Gateway

- **Role:** Processes financial transactions between clients and freelancers once payment features are implemented. Ensures secure, transparent, and reliable money transfers.
- **Key Interaction:** Handles job payments, refunds (if needed), transaction logs, and payment confirmations for both parties.

User Profile

User Profile-01: Client (Job Poster / Buyer)

Characteristic	Notes on characteristic	Requirement implied
Type of user	Client (Job Poster / Buyer)	Verification
Age range	18-60	Performance, Operation
Frequency of use	Uses the system whenever needing to hire a freelancer	Performance, Scalability, Acceptance

Mandatory	Yes	Security, Operation
Computer experience	Varies (basic to advanced)	Documentation
Education	Varies (students, professionals, business owners)	Resource, Performance, Operation, Acceptance
Goals	Post jobs, review bids, hire freelancers, manage ongoing tasks	Resource, Performance, Security, Acceptance, Operation, Portability
Language skills	Bangla, English	Accessibility, Acceptance
Number of users	30000+	Performance, Operation, Acceptance, Scalability, Security
Training	Minimal training needed	Documentation
Other systems used	Email, workplace tools	Integration
Ways of working	Creates job listings, reviews proposals, selects freelancers	Creates job listings, reviews proposals, selects freelancers

User Profile-02: Freelancer (Service Provider / Job Seeker)

Characteristic	Notes on characteristic	Requirement implied
Type of user	Freelancer (job seeker)	Verification
Age range	18-60+	Performance, Operation
Frequency of use	Frequently checks jobs, applies regularly	Performance, Operation, Security, Reliability
Mandatory	Yes	Security, Operation

Computer experience	Intermediate (knows web platforms)	Documentation
Education	Skill-based; varies from beginner to expert	Resource, Performance
Goals	Search jobs, submit bids, secure work, track bid status	Operation, Acceptance
Language skills	English, Bangla	Accessibility
Number of users	30,000+ freelancers	Scalability, Security
Training	Minimal guidance required	Documentation
Other systems used	Portfolio websites, communication tools	Integration
Ways of working	Browses jobs, submits proposals, communicates with client	Operation, Security

User Profile-03: System Administrator

Characteristic	Notes on characteristic	Requirement implied
Type of user	System Administrator	Verification
Age range	25-50+	Performance, Reliability
Frequency of use	Regularly	Performance, Security
Mandatory	Yes	Security, Operation

Computer experience	High	Documentation
Education	Relevant IT/Software background	Resource, Security,
Goals	Manage system stability, user accounts, security, logs	Operation, Maintenance
Language skills	English, Bangla	Acceptance
Number of users	Limited (5–20 admins)	Security
Training	Essential	Documentation
Other systems used	Monitoring tools, admin dashboards	Integration
Ways of working	Oversees system operations, resolves disputes, maintains data	Operation, Safety, Maintenance, Portability

User Profile-04: Notification & System Services (Internal)

Characteristic	Notes on characteristic	Requirement implied
Type of user	Internal service module	Verification
Age range	Not applicable	Performance
Frequency of use	Continuous background operation	Scalability, Reliability
Mandatory	Yes	Security
Computer experience	Automated	Documentation
Education	Not applicable	Resource

Goals	Send alerts, update bid status, notify users	Operation
Language skills	System-based	Portability
Number of users	Not a person (system process)	Integration
Training	Not applicable	Maintenance
Other systems used	Database + front-end event triggers	Integration
Ways of working	Triggers notifications on bid updates/job changes	Operation, Security

Scope

1. User Registration and Secure Login

- **Objective:** Allow clients and freelancers to create accounts and securely access the system.
- **Functionality:**
 - A simple and secure registration process requiring basic user information.
 - Authentication system ensuring only verified users can access job-related features.
 - Password recovery options for easy account restoration.

2. Job Posting and Management

- **Objective:** Enable clients to create, update, and manage job listings.
- **Functionality:**
 - Clients can post new jobs with details such as title, description, budget, and category.
 - Ability to edit or delete job postings at any time.
 - Posted jobs are stored in the system and visible to freelancers for bidding.

3. Job Browsing and Search

- **Objective:** Allow freelancers to find suitable jobs efficiently.
- **Functionality:**
 - Browse through all available job listings.
 - Search using keywords and apply category-based filters.
 - View detailed job descriptions before submitting proposals.

4. Bid Submission System

- **Objective:** Allow freelancers to apply for jobs with customized proposals.
- **Functionality:**
 - Freelancers can submit bids including price, proposal message, and delivery time.
 - System prevents duplicate bids by the same user on the same job.
 - Each bid increments the job's bid count automatically.

5. Bid Review and Decision

- **Objective:** Help clients evaluate proposals and choose the best freelancer.
- **Functionality:**
 - Clients can view all bids submitted for their posted jobs.
 - Ability to approve or reject bids based on freelancer details and proposal quality.
 - Updates are reflected instantly for both parties.

6. My Bids and My Posted Jobs

- **Objective:** Allow users to track their activities in the system.
- **Functionality:**
 - Freelancers can track all bids they have submitted and see their status.
 - Clients can view and manage all jobs they have posted.
 - Activity summaries displayed on personalized dashboards.

7. Job Update and Deletion

- **Objective:** Provide clients flexibility in managing their job listings.
- **Functionality:**
 - Users can update job details using an editable form.
 - Jobs can be deleted when they are no longer needed.

- o Changes are stored in the database and updated in real-time.

8. Notification and Status Updates

- **Objective:** Ensure users remain informed about important actions.
- **Functionality:**
 - o Freelancers receive notifications when their bids are accepted or rejected.
 - o Clients receive alerts when new bids are submitted.
 - o Updates displayed clearly on dashboards for easy tracking.

9. Secure and Reliable Data Handling

- **Objective:** Protect user information and ensure smooth system operation.
- **Functionality:**
 - o Authentication and role-based access to protect sensitive data.
 - o Secure communication with the backend using API endpoints.
 - o Regular updates and structured backend architecture (Node.js + MongoDB).

10 Future Payment & Communication Integration (Planned Scope)

- **Objective:** Extend the platform to support complete project lifecycle management.
- **Functionality:**
 - o Secure payment processing between client and freelancer.
 - o Direct messaging or live chat for better communication.
 - o Rating & review system after job completion.

Feasibility Study

This feasibility study evaluates the practicality of developing and implementing the SoloSphere platform by analyzing its technical, operational, economic, and scheduling aspects.

1. Technical Feasibility

a. Hardware Compatibility

- The system is fully web-based and can run on desktops, laptops, tablets, and smartphones.
- Since it uses cloud-hosted backend services (MongoDB Atlas, Node.js server), no heavy hardware is required from users.
- Cross-device responsive UI ensures accessibility for both clients and freelancers.

b. Software Compatibility

- The platform is developed using React (Frontend), Node.js/Express (Backend), and MongoDB Atlas (Database).
- Compatible with all major operating systems including Windows, macOS, Linux, Android, and iOS.
- REST API architecture makes it easy to integrate additional services like payment gateways and messaging systems in the future.

c. Technical Expertise

- Development requires knowledge of modern web technologies such as JavaScript, React, Node.js, Express, and database management.
- Admins may need basic training to manage user accounts, monitor jobs, and handle disputes.
- The system is easy to maintain due to modular folder structure and clean REST APIs.

2. Operational Feasibility

a. User Acceptance

- The system provides an intuitive, user-friendly interface for posting jobs, submitting bids, and managing workflows.
- Both clients and freelancers can easily adapt to the platform due to simple navigation and clear workflows.
- Early user feedback can be used to refine UX/UI elements.

b. Impact on Current Operations

- SoloSphere reduces communication gaps between clients and freelancers by centralizing job posting, bidding, and decision-making.

- Real-time updates, bid tracking, and job management streamline the overall hiring process.
- Support staff and admins can efficiently monitor system activities without heavy workload.

3. Economic Feasibility

a. Cost-Benefit Analysis

- **Development Costs:** Backend server setup, database hosting, domain, security, and developer effort..
- **Operational Costs:** Cloud database storage (MongoDB Atlas), backend hosting (Render/Heroku/VPS), maintenance, updates, and customer support.
- **Benefits:** Automated freelance hiring, reduced communication delays, and improved project completion efficiency for both clients and freelancers.

b. Return on Investment (ROI)

- Long-term revenue generation through freelancer commissions, ads, and premium features.
- Reduced operational overhead due to automated bidding and hiring processes.

4. Scheduling Feasibility

a. Project Timeline

The implementation will follow these phases:

1. **Requirement Analysis & Planning (1 months)** – Identify user needs, finalize architecture, set development goals.
2. **Development & Testing (2-4 months)** – Build job posting, bidding, user dashboards, and authentication systems.
3. **Pilot Testing (1 months)** – Functional testing, bug fixing, performance improvement.
4. **Deployment & Pilot Testing (1 month)** – Deploy test version, gather feedback, refine UI/UX.
5. **Full Implementation & Maintenance (Ongoing)** – Roll out finished product with monitoring and updates.

b. Dependencies

- Stable internet connectivity and API reliability.
- Integration readiness for future payment systems.

- Availability of developer resources for continuous improvements.

5. Security and Privacy Considerations

a. Data Encryption

- Secure handling of user credentials and sensitive information.
- All communication between client and server uses HTTPS encryption.
- Passwords stored securely using authentication services (Firebase/Auth/JWT as applicable).

b. Compliance

- Role-based permissions prevent unauthorized access.
- Secure CRUD operations for jobs and bids.
- Protection against common vulnerabilities like SQL/NoSQL injection, XSS, and CSRF.

The feasibility study indicates that SoloSphere is technically, operationally, economically, and schedule-wise fully viable.

With modern technologies, scalable architecture, and user-centered design, the system can reliably support a freelancer marketplace and evolve into a full-featured hiring ecosystem.

Functional Requirements

FR01	Registration
Description	Users must be able to register an account by providing necessary information such as name, email, and password.
Stakeholder	Client, Freelancer

FR02	Login
------	--------------

Description	Registered users must be able to securely log in to access the platform.
Stakeholder	Client, Freelancer

FR03	Google Login
Description	The system must allow authentication via Google Sign-In.
Stakeholder	Client, Freelancer

FR04	Secure Private Routes
Description	Pages such as Add Job, My Posted Jobs, My Bids, and Bid Requests must be accessible only to logged-in users..
Stakeholder	Client, Freelancer

FR05	Home Page Display
Description	The home page must show a banner, tab category section, and job cards fetched from the database.
Stakeholder	All user

FR06	Job Posting
Description	Clients can create new job posts by providing title, category, deadline, description, and salary range.
Stakeholder	Client

FR07	View All Jobs
Description	Users must be able to view all job listings on the All Jobs page.
Stakeholder	Client, Freelancer

FR08	Search Jobs
Description	Users can search jobs by title through a search field.
Stakeholder	Client, Freelancer

FR09	Filter Jobs by Category
Description	The system must filter job listings based on selected categories/tabs.
Stakeholder	Freelancer

FR10	Job Details View
Description	Users can view full job details including title, category, deadline, description, salary range, and buyer information.
Stakeholder	Client, Freelancer

FR11	Update Job
Description	Job owners must be able to update any job they previously posted.
Stakeholder	Client

FR12	Delete Job
Description	Job owners must be able to delete a job from the system.
Stakeholder	Client

FR13	Apply/Bid on a Job
Description	Freelancers can place a bid by submitting a proposal, expected delivery date, and asking price.
Stakeholder	Freelancer

FR14	Increment Bid Count
Description	The system must automatically increase the job's bid_count when a new bid is placed.
Stakeholder	Freelancer, Client

FR15	Prevent Duplicate Bids
Description	A freelancer must not be allowed to bid more than once on the same job.
Stakeholder	Freelancer

FR16	View My Bids
Description	Freelancers must be able to view all bids they have submitted along with bid status.
Stakeholder	Freelancer

FR17	Bid Requests for Client
Description	Clients must see all bids submitted for their posted jobs on the "Bid Requests" page.
Stakeholder	Client

FR18	Update Bid Status
Description	Clients can accept or reject bids, and the system must update the bid's status instantly.
Stakeholder	Client

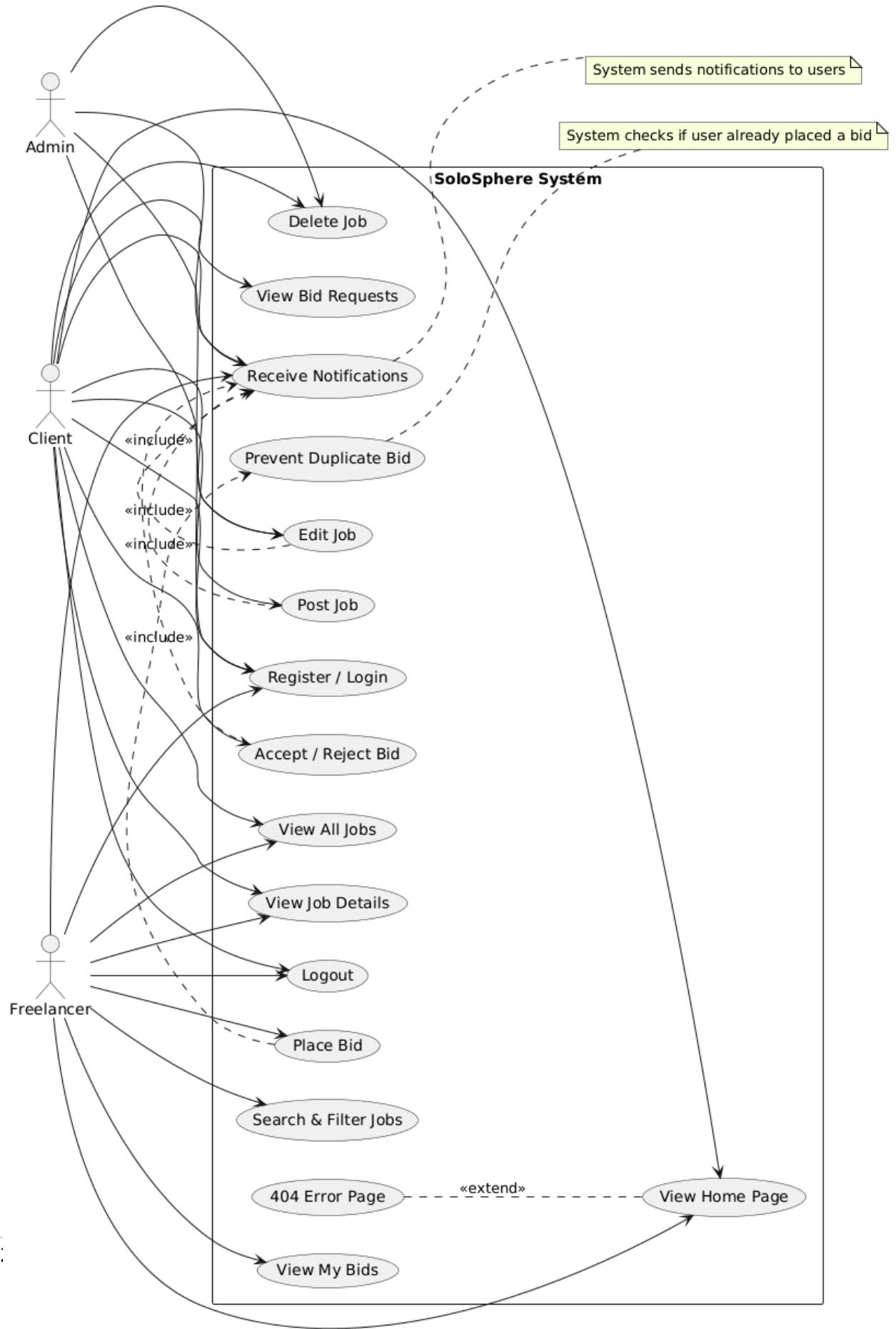
FR19	Deadline Restriction (Optional Feature)
Description	Freelancers must not be able to apply for jobs with expired deadlines.
Stakeholder	Freelancer

FR20	Error/404 Page
Description	A user-friendly error page must display when navigating to an invalid route..
Stakeholder	All user

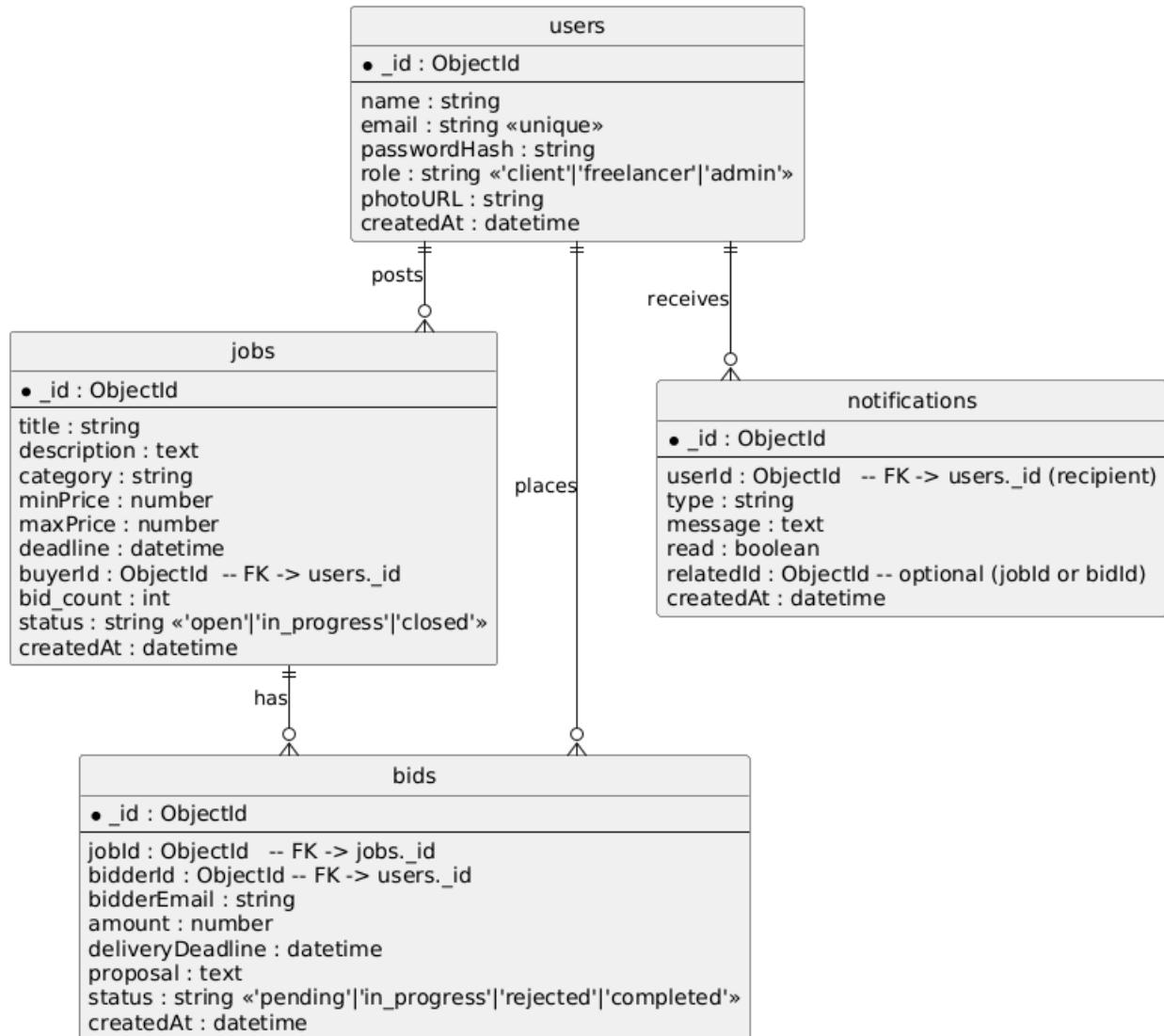
FR21	Secure API Interaction
Description	Frontend must communicate with backend API using provided environment variables (.env.local).
Stakeholder	System, Developer

FR22	Log Out
Description	Users must be able to securely log out from their account.
Stakeholder	Client, Freelancer

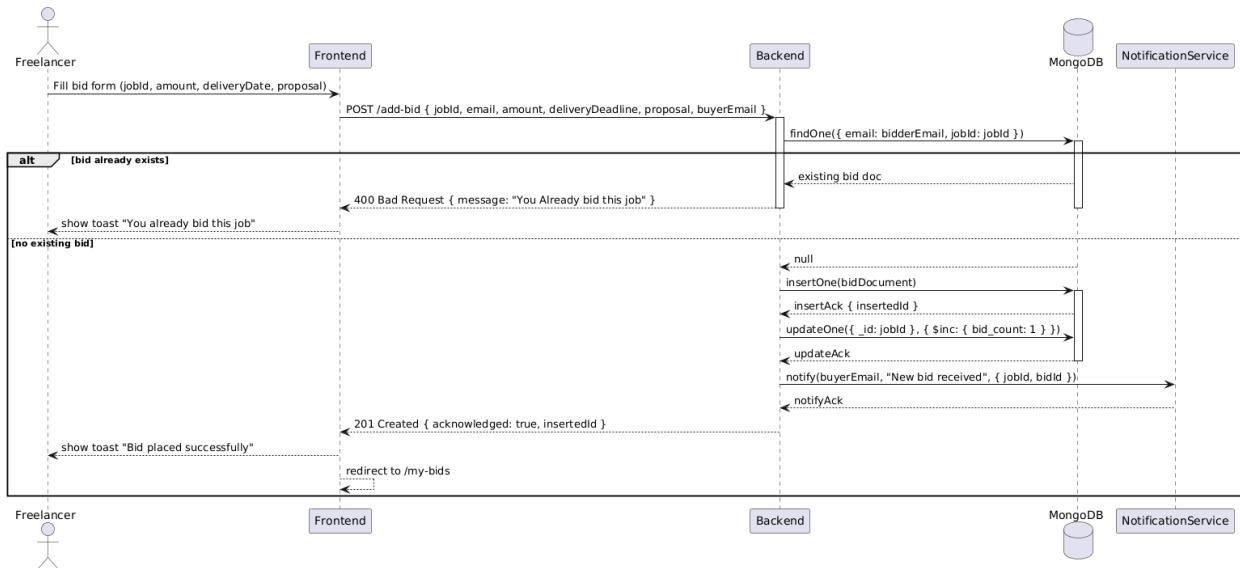
Use Case Diagram



ER Diagram



Sequence Diagram



Activity Diagram

